Rajalakshmi Engineering College

Name: Manishraj A
Email: 240801192@rajalakshmi.edu.in
Roll no: 240801192
Phone: 9514051184
Branch: REC
Department: I ECE AF
Batch: 2028
Degree: B.E - ECE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Imagine you are working on a text processing tool and need to implement a feature that allows users to insert characters at a specific position.

Implement a program that takes user inputs to create a singly linked list of characters and inserts a new character after a given index in the list.

Input Format

The first line of input consists of an integer N, representing the number of characters in the linked list.

The second line consists of a sequence of N characters, representing the linked list.

The third line consists of an integer index, representing the index(0-based) after which the new character node needs to be inserted.

The fourth line consists of a character value representing the character to be inserted after the given index.

Output Format

If the provided index is out of bounds (larger than the list size):

1. The first line of output prints "Invalid index".
2. The second line prints "Updated list: " followed by the unchanged linked list values.

Otherwise, the output prints "Updated list: " followed by the updated linked list after inserting the new character after the given index.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
a b c d e
2
X
Output: Updated list: a b c X d e

Answer

```
#include <stdio.h>
#include <stdlib.h>
struct node{
    char data;
    struct node *next;
}*head=NULL;
typedef struct node node;
void insert(char value){
    struct node* newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=value;
    newnode->next=NULL;
    if(head==NULL)
    head=newnode;
    else{
        node *temp=head;
        while(temp->next!=NULL)
        temp=temp->next;
        temp->next=newnode;
    }
}

void inbeg(char value){
    if(head==NULL)
    return;
    node*newnode=(node*)malloc(sizeof(node));
    newnode->data=value;
    newnode->next=head->next;
    head->next=newnode;
}
void last(char value){
    if(head==NULL)
    return;
    node *pos=head;
    while(pos->next!=NULL)
    pos=pos->next;
    node* newnode=(node*)malloc(sizeof(node));
    newnode->data=value;
    newnode->next=pos->next;
    pos->next=newnode;
}
void mid(int pos,char value){
    node* temp=head;
    int count=0;
    while(temp!=NULL && count<pos)
    {
        temp=temp->next;
        count++;
    }
    if(temp==NULL)
    return;
    else{
        node* newnode=(node*)malloc(sizeof(node));
        newnode->data=value;
        newnode->next=temp->next;
        temp->next=newnode;
    }
}

void display()
{
    node *temp=head;
    while(temp!=NULL)
    {
        printf("%c ",temp->data);
        temp=temp->next;
    }
    printf("\n");
}

int main(){
    int n,pos;
    char value,a;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf(" %c",&value);
        insert(value);
    }
    scanf("%d",&pos);
    scanf(" %c",&a);
    if(pos<0||pos>n)
    {
        printf("Invalid index\n");
        printf("Updated list: ");
        display();
    }
    else if(pos==0)
    {
        inbeg(a);
        printf("Updated list: ");
        display();

    }
    else if(pos==n-1)
    {
        last(a);
        printf("Updated list: ");
        display();
    }
    else{
        mid(pos,a);
        printf("Updated list: ");
        display();
    }
}
```

Status : Correct *Marks : 10/10*